

UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTT	PPPPPPPPPPPPPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEE	TTT	PPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPP
UUU	UUU	EEEEEEEEEEEE	TTT	PPP

\*\*FILE\*\*ID\*\*RMSTEST1

M 4

RM  
VO

RRRRRRRRR      MM      MM      SSSSSSSSS      TTTTTTTTTTT      EEEEEEEEEE      SSSSSSSSS      TTTTTTTTTTT      11  
RRRRRRRRR      MM      MM      SSSSSSSSS      TTTTTTTTTTT      EEEEEEEEEE      SSSSSSSSS      TTTTTTTTTTT      11  
RR      RR      MMMMM      MMMMM      SS      TT      EE      SS      TT      1111  
RR      RR      MMMMM      MMMMM      SS      TT      EE      SS      TT      1111  
RR      RR      MM      MM      MM      TT      EE      SS      TT      1111  
RR      RR      MM      MM      MM      TT      EE      SS      TT      1111  
RRRRRRRRR      MM      MM      SSSSSSS      TT      EEEEEEEE      SSSSSSS      TT      1111  
RRRRRRRRR      MM      MM      SSSSSSS      TT      EEEEEEEE      SSSSSSS      TT      1111  
RR      RR      MM      MM      SS      TT      EE      SS      TT      1111  
RR      RR      MM      MM      SS      TT      EE      SS      TT      1111  
RR      RR      MM      MM      SS      TT      EE      SS      TT      1111  
RR      RR      MM      MM      SSSSSSSSS      TT      EEEEEEEEEE      SSSSSSSSS      TT      111111  
RR      RR      MM      MM      SSSSSSSSS      TT      EEEEEEEEEE      SSSSSSSSS      TT      111111

```
0000 64      $BEGIN RMSTEST1.009,--RMSTEST,<GENERAL RMS TEST PROGRAM>,<GBL,LONG>
0000 65
0000 66 :
0000 67
0000 68      .ENABL DBG
0000 69
0000 70 :
0000 71 : this program tests the sequential file org. for a disk device.
0000 72 : with vfc record format.
0000 73
0000 74 :
0000 75 : test 1a:
0000 76 :
0000 77 : create known test file of 1000 records, where each record has
0000 78 : a 4-byte fixed control field giving the record # and a variable
0000 79 : length portion of n mod 100 bytes of the ascii character n mod 10
0000 80 : (where 'n' is the record number).
0000 81 :
0000 82
0000 83      $RMSDEF
0000 84      .NLIST MEB
0000 85
0000 86 :
0000 87 : macros:
0000 88 :
0000 89
0000 90      .MACRO BUFF NAM,SIZE
0000 91 NAM'BUF:::
0000 92      .BLKB SIZE
0000 93      NAM'BSZ==SIZE
0000 94      .ENDM BUFF
0000 95
0000 96 :
0000 97 :
0000 98 :
0000 99
0000 100     .MACRO TYPE STRING, ?L
0000 101     STORE <STRING>
0000 102     BLBC VERTOSITY,L
0000 103     MOVL #SS.TMPX,CMDORAB+RABSL,RBF
0000 104     MOVW #SS.TMPX1,CMDORAB+RABSO,RSZ
0000 105     SPUT RAB=CMDORAB,ERR=REPORT_ERROR
0000 106     BSBW ERR
0000 107 L:   .ENDM TYPE
0000 108
0000 109 :
0000 110 :
0000 111
0000 112     .MACRO WTYPE STRING
0000 113     SWAIT RAB=CMDORAB
0000 114     TYPE <STRING>
0000 115     .ENDM WTYPE
0000 116
0000 117     .MACRO WFIELD STRING
0000 118     SWAIT RAB=CMDORAB
0000 119     FIELD <STRING>
0000 120     .ENDM
```

RMSTEST1  
V04-000

GENERAL RMS TEST PROGRAM :

B 5

16-SEP-1984 01:45:37 VAX/VMS Macro V04-00  
5-SEP-1984 04:21:39 [UETP.SRC]RMSTEST1.MAR;1

Page 2  
(1)

0000 121  
0000 122 :  
0000 123

RM  
VO

```
0000 125 .MACRO STORE STRING,PRE
0000 126 .SAVE
0000 127 .PSECT $RMSNAM
0000 128 $$TMPX=-
0000 129 PRE
0000 130 ASCII %STRING%
0000 131 $$TMPX1=-$$TMPX
0000 132 .RESTORE
0000 133 .ENDM STORE
0000 134
0000 135 :
0000 136
0000 137 .MACRO BEGIN TSTNAM
0000 138 STORE <TSTNAM>
0000 139 MOVL #$$TMPX,BEG_DESCR+4 : addr
0000 140 MOVL #$$TMPX1,BEG_DESCR : len
0000 141 BSBW BEGPUT
0000 142 .ENDM BEGIN
0000 143 .MACRO FINISH TSTNAM
0000 144 STORE <TSTNAM>
0000 145 MOVL #$$TMPX,FIN_DESCR+4 : addr
0000 146 MOVL #$$TMPX1,FIN_DESCR : len
0000 147 BSBW FINPUT
0000 148 .ENDM FINISH
0000 149 .MACRO FIELD FLDNAM
0000 150 STORE <FLDNAM>
0000 151 MOVL #$$TMPX,FLD_DESCR+4 : addr
0000 152 MOVL #$$TMPX1,FLD_DESCR : len
0000 153 BSBW FLDPUT
0000 154 .ENDM FIELD
0000 155 .MACRO MBPT, ?L
0000 156 BLBC VERBOSITY,L
0000 157 BPT
0000 158 L:
0000 159 .ENDM MBPT
0000 160
0000 161 :
0000 162
```

```

00000000 164 .PSECT RMSTEST.GBL, LONG
0000 165 .ALIGN 10NG
0000 166 T1START:::
0000 167 WTRAB::: $RAB
0000 168 RFATBL: .BLKQ 100
0000 169 T1FAB::: $FAB FAC=PUT, FNM=<TSTS DISK:T1FILE.DAT;1>, org=seq, rfm=vfc, -
0000 170 RAT=CR, FSZ=4, MRS=100, NAM=NAMBLK, FOP=<SUP, CTG>,-
0000 171 ALQ=48, DEQ=12, SHR=<PUT, GET, UPI>
0000 172 T1RAB::: $RAB FAB=T1FAB, UBF=CPYBUF, USZ=CPYBSZ, RBF=CPYBUF, MBC=4, MBF=2,-
0000 173 ROP=<WBH>, RHB=RECCNT, KBF=RECCNT
00000000 00000000 03F8 RECCNT:::
00000000 00000000 03F8 174 .LONG 0,0
00000000 00000000 0400 175 :
00000000 00000000 0400 176 :
00000000 00000000 0400 177 :
00000000 00000000 0400 178 :
00000000 00000000 0400 179 T1STR: .LONG T1L, T1S
00000000 00000000 0408 180 T1S: .ASCII '!4UL. RECORDS WRITTEN. RFA = !XL,!XW'
44 52 4F 43 45 52 20 2E 4C 55 34 21 0408
20 20 2E 4E 45 54 54 49 52 57 20 53 0414
58 21 2C 4C 58 21 20 3D 20 41 46 52 0420
57 042C
00000025 042D 181 T1L=-T1S
042D 182
042D 183 :
042D 184
00000435 042D 185 T2STR: .LONG T2L, T2S
00000005F 0435 186 T2S: .ASCII 'RECORD # = !4UL, RHB = !4UL!/'
20 3D 20 23 20 44 52 4F 43 45 52 20 0441
20 3D 20 42 48 52 20 2C 4C 55 34 21 044D
2F 21 4C 55 34 21 0453
4C 55 33 21 20 3D 20 45 5A 49 53 20 045F
21 57 55 21 20 3D 20 5A 53 52 20 2C 046B
2F 046C
3D 20 41 46 52 20 44 45 56 41 53 20 0478
49 46 20 2C 57 58 21 2C 4C 58 21 20 0484
4C 58 21 20 3D 20 41 46 52 20 45 4C 0490
57 58 21 2C 0494
0000005F 0494 189 T2L=-T2S
0494 190
0494 191 :
0494 192
0000049C 0494 193 T3STR: .LONG T3L, T3S
00000000F 049C 194 T3S: .ASCII 'RECORD = '!AF'!/'"
2F 21 27 04A8
00000000F 04AB 195 T3L=-T3S
01 04AB 196 RHBSW: .BYTE 1

```

; switch for modifying rhb contents

FB36 CF 00000002'EF B0 04C3 198 RMTTEST\_1A::  
OFFC 04AC 199 WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>  
04AE 200 BEGIN <SEQUENTIAL TESTS>  
04CC 201 MOVW CMDORAB+RAB\$W\_ISI,WTRAB+RAB\$W\_ISI: copy output isi for  
04CC 202  
04CC 203 : DIFFERENT RAB SWAIT TEST  
04CC 204  
04CC 205 :  
04CC 206  
FE93 CF 00100000 8F CA 04CC 207 BICL2 #FABSM\_CTG,T1FAB+FABSL\_FOP; don't want ctg set  
00000004'EF 01 C8 04D5 208 BISL2 #RABSM\_ASY,CMDORAB+RAB\$L\_ROP  
FF17 CF 01 D0 04DC 209 MOVL #1,RECCNT  
58 FB5F CF DE 04E1 210 MOVAL RFATBL,R8  
FB06' 30 04F7 211 \$CREATE FAB=T1FAB,ERR=REPORT\_ERROR  
5B FEB6 CF DE 04FA 212 BSBW ERR  
FAEF' 30 050E 213 MOVAL T1RAB,R11  
0511 214 \$CONNECT RAB=R11,ERR=REPORT\_ERROR  
0511 215 BSBW ERR  
0511 216 :  
0511 217 :  
52 FEDE CF 00000064 8F 7B 0511 218 NXTREC: EDIV #100,RECCNT,R2,R6 : compute record length  
56 051B  
57 52 FED7 CF 0A 7B 051C 220 EDIV #10,RECCNT,R2,R7 : compute character for record  
57 30 80 0523 221 ADDB #48,R7 : make it ascii  
22 AB 56 B0 0526 222 MOVW R6,RAB\$W\_RSZ(R11) : tell rms the record size  
56 57 6E 00 2C 052A 223 MOVC5 #0,(SP),R7,R6,CPYBUF : fill record with character  
00000000'EF 052F  
FABA' 30 0534 224 SPUT RAB=R11,ERR=REPORT\_ERROR  
30 57 D1 0543 225 BSBW ERR  
0B 12 0546 226 CMPL R7,#48 : is this 10th record?  
88 10 AB 7D 0548 227 BNEQ T1CNT : branch if not  
56 D5 054F 228 MOVQ RAB\$W\_RFA(R11),(R8)+ : save rfa in table  
03 12 0551 229 TSTL R6 : is this 100th record?  
093F 30 0553 230 BNEQ T1CNT  
0556 231 BSBW TYPRFA : type out the rfa  
0556 232 :  
0556 233 :  
0556 234 :  
FFB1 FE9A CF 01 03E8 8F 3D 0556 235 T1CNT: ACBW #1000,#1,RECCNT,NXTREC  
FA8C' 30 0560 236 \$CLOSE FAB=T1FAB,ERR=REPORT\_ERROR  
FE3E CF 84 0571 237 BSBW ERR  
FA8C' 30 0574 238 CLRW T1RAB+RAB\$W\_ISI : allow rab to be re-used  
0578 239 SWAIT CMDORAB  
0585 240 TYPE <FINISHED TEST 1A - CREATED TEST FILE>

```

05B4 242
05B4 243
05B4 244 ; test 1b
05B4 245
05B4 246 ; re-read file created in test 1a and try random and sequential
05B4 247 ; access via $get and $find.
05B4 248
05B4 249
05B4 250 ; $WAIT RAB=CMDORAB
05C1 251 ; TYPE <BEGINNING TEST 1B - $GET AND $IND>
05F0 252 ; BISL2 #FAB$M_NAM,T1FAB+FAB$L_FOP
05F9 253 ; BISB2 #FAB$M-GET,T1FAB+FAB$B-FAC
05FE 254 ; $OPEN FAB=T1FAB,ERR=REPORT_ERROR
060F 255 ; BSBW ERR

FD6F CF 01000000 8F C8 05F0 256
FD7C CF 02 88 05F9 257
F9EE' 30 060F 258 ; \eventually include code here to verify file attributes.\

04 AB 00010000 8F D0 0612 259
0612 260 ; MOVL #RAB$M_LOC,RAB$L_ROP(R11); locate mode
061A 261 ; $CONNECT RAB=T1RAB,ERR=REPORT_ERROR
062B 262 ; BSBW ERR
062E 263 ; MOVB #RAB$C_RFA,RAB$B_RAC(R11); rfa access
0632 264 ; MOVZWL #1000,R9 ; starting record number
0637 265 ; CLRL R10 ; as a quadword for ediv
0639 266 ; CLRL R7 ; flag for 1st pass
063B 267
063B 268
063B 269
063B 270 ; T1BL0OP1:
063B 271 ; SUBL #8,R8 ; move to previous rfa
063E 272 ; BSBW GETANDCHK
0641 273 ; ACBW #10,-#10,R9,T1BL0OP1
0649 274 ; MOVL #10,R9 ; start with record # 10
064C 275
064C 276
064C 277 ; T1BL0OP2:
064C 278 ; BSBW GETANDCHK
064C 279 ; ADDL #8,R8 ; move to next rfa
064F 280 ; ACBW #990,#10,R9,T1BL0OP2
0652 281
065A 282
065A 283
065A 284 ; T1BDONE:
065A 285 ; BBSS #0,R7,T1BDONE ; branch if 2nd pass
065E 286 ; SWAIT CMDORAB
066B 287 ; TYPE <PASS 1 O.K. >
069A 288 ; MOVZWL #990,R9 ; start with rec # 990
069F 289 ; BRB T1BL0OP1
06A1 290
06A1 291 ; SWAIT CMDORAB
06AE 292 ; TYPE <PASS 2 O.K. >
06DD 293 ; SCLOSE FAB=T1FAB,ERR=REPORT_ERROR
F90F' 30 06EE 294 ; BSBW ERR

```

```

06F1 297
06F1 298 : test 1c
06F1 299 : use update to modify records created in test 1a
06F1 300
06F1 301
06F1 302
06F1 303
06F1 304 WTTYPE <START TEST 1C - SUPDATE>
072D 305 BISB2 #FAB$M_UPD,T1FAB+FAB$B FAC
0732 306 $OPEN FAB=T1FAB,ERR=REPORT_ERROR
0743 307 BSBW ERR
0746 308 CLRW RABSW_ISI(R11)
0749 309 $CONNECT RAB=R11,ERR=REPORT_ERROR
0758 310 BSBW ERR
075B 311 MOVB #RAB$C_RFA,RAB$B_RAC(R11)
075F 312 MOVL #10,R9 ; start at record 10
0762 313 MOVAL RFATBL,R8
0767 314 CLRL R7 ; get single record
0769 315
0769 316 :
0769 317
0769 318 T1BLOOP4:
0769 319 BSBW GETANDCHK
076C 320 INCL RECCNT ; modify rhb
0770 321 MNEG B RHBSW,RHBSW ; toggle rhb modify flag
0777 322 BLSS 10$ ; don't modify rhb (default)
0779 323 CLRL RAB$L_RHB(R11) ; get record addr
077C 324 10$: MOVL RAB$L_RBF(R11),R5 ; and len
0780 325 MOVZWL RAB$W_RSZ(R11),R4 ; modify record contents
0784 326 20$: INC B (R5)+ R4,20$ ; restore rhb addr
0786 327 SOBGTR RAB=R11,ERR=REPORT_ERROR
0789 328 BSBW ERR
0798 329 MOVAL RECCNT,RAB$L_RHB(R11) ; bump to next rfa entry
079B 330 ADDL #8,R8 ; modify every 10th record
07A1 331 ACBW #1000,#10,R9,T1BLOOP4
07A4 332
07AC 333
07AC 334 : now reread the modified file and check that every 10th (and only
07AC 335 : every 10th) record has been correctly modified
07AC 336
07AC 337
07AC 338
07AC 339 MOVAL RFATBL,R8
0781 340 CLRL R9 ; reset record #
0783 341 MOV B #RAB$C_SEQ,RAB$B_RAC(R11)
0787 342
0787 343 : $disconnect r11
0787 344 : bsbw err
0787 345 : $connect r11 ; do an effective rewind
0787 346
0787 347
0787 348
0787 349
0787 350 F837' 30 07C6 351 $REWIND RAB=R11,ERR=REPORT_ERROR; so do a rewind
07C9 352 BSBW ERR
07C9 353 :

```

```

07C9 354
07C9 355 T1BLOOP5:
57 D6 07C9 356 INCL R7 ; set switch for no rfa compare
09 DD 07CB 357 PUSHL #9 ; # gets of unmodified recs
59 D6 07CD 358 10$: INCL R9 ; bump rec #
F81F' 30 07DE 359 $GET RAB=R11,ERR=REPORT_ERROR
057A 30 07E1 360 BSBW ERR
E6 6E F5 07E4 361 BSBW CHKREC
8E D5 07E7 362 SOBGTR (SP),10$ ; bump rec #
59 D6 07E9 363 TSTL (SP)+ ; specify rfa to be checked
07EB 364 INCL R9 ; toggle rhb modified switch
F803' 30 07FA 365 $GET RAB=R11,ERR=REPORT_ERROR
57 D4 07FD 366 BSBW ERR
FC45 CF FC48 CF 04 19 0806 368 MNEGB RHBSW,RHBSW
04 19 0806 369 BLSS 15$ ; toggle rhb modified switch
50 FBE7 CF 59 C3 080C 370 INCL RECCNT
50 D7 0812 371 15$: SUBL3 R9,RECCNT,RO ; rhb contents = rec # + 1?
03 13 0814 372 DECL RO
0590 30 0816 373 BEQL 20$ ; branch if yes
0549 30 0819 374 BSBW BADRHB ; report error
03E8 58 08 C0 081C 376 BSBW CHKRC1 ; check record len, rfa, and contents
BF 59 B1 081F 377 ADDL #8,R8 ; bump rfa table addr
A3 12 0824 378 CMPW R9,#1000 ; done?
0826 379 BNEQ T1BLOOP5 ; branch if not
F7C6' 30 0837 380 SCLOSE FAB=T1FAB,ERR=REPORT_ERROR
BSBW ERR

```

083A 382  
 083A 383  
 083A 384 : test 2 - random i/o test for sequential file org  
 083A 385  
 083A 386  
 083A 387  
 083A 388  
 083A 389  
 083A 390 : create a known test file of 1000 records of fixed length = 49. bytes.  
 083A 391 : 1st longword has the record # (n) followed by 45 bytes of the ascii  
 083A 392 : character (n mod 42) + 48.  
 083A 393 :  
 083A 394 : WTTYPE <START TEST 2A - RANDOM SEQ I/O>  
 T2SETUP:  
 28 AB 02 AB B4 0876 395 : CLRW RAB\$W ISI(R11)  
 00000000'FF DE 0879 396 : MOVAL CPYBUF, RAB\$L RBF(R11)  
 22 AB 31 9B 0881 397 : MOVZBH #49, RAB\$W\_RSZ(R11)  
 5A FADB CF DE 0885 398 : MOVAL T1FAB, R10  
 10 AA 30 9A 088A 399 : MOVZBL #48, FABSL ALQ(R10)  
 04 AA 01 90 0899 400 : BICL2 #FABSM\_NAM, FABSL\_FOP(R10)  
 16 AA 01 94 0896 401 : CLRBL FABSB FAC(R10) : check for put default  
 1F AA 01 90 0899 402 : MOVB #FABSC FIX, FABSB RFM(R10)  
 36 AA 31 B0 089D 403 : MOVW #49, FAB\$W\_MRS(R10) : rec len  
 08A1 404 : SRAB STORE RAB=(R11), ROP=<LOC,UIF>  
 08A9 405 : SCREATE FAB=R10, ERR=REPORT\_ERROR  
 BSBW 406 :  
 FAAB CF 00010631 8F D1 0888 407 : CMPL #RMSS\_SUPERSEDE, T1FAB+FABSL\_STS  
 15 13 08C4 408 : BEQL SUPOK  
 F745' 30 08DB 409 : FIELD <STATUS WORD IS NOT SUPERSEDE, THEREFORE IT>  
 F713' 30 08EA 410 : SCONNECT RAB=R11, ERR=REPORT\_ERROR  
 BSBW 411 :  
 08ED 412 :  
 08ED 413 : pre-extend file on 1st pass, put sequentially on second pass  
 08ED 414 :  
 08ED 415 :  
 08ED 416 :  
 FAFF CF 1F 1E AA 03 E0 08ED 417 : BBS #FAB\$V\_BLK, FABSB\_RAT(R10), 10\$; branch if pass 2  
 03E8 8F 3C 08F2 418 : MOVZWL #1000, RECCNT  
 1E AB 01 90 08F9 419 : MOVB #RABSC KEY, RABSB RAC(R11)  
 F6F1' 30 090C 420 : SPUT RAB=R11, ERR=REPORT\_ERROR  
 04 11 090F 421 : BSBW ERR  
 1E AB 00 90 0911 422 : BRB 20\$  
 FADE CF 01 3C 0915 423 : 10\$: MOVBL #RABSC SEQ, RABSB\_RAC(R11)  
 091A 424 : 20\$: MOVZWL #1, RECCNT  
 091A 425 :  
 091A 426 :  
 091A 427 :  
 57 52 FAD9 CF 2A 78 091A 428 : NXTRC2: EDIV #42, RECCNT, R2, R7 : compute char for record  
 57 30 80 0921 429 : ADDB2 #48, R7 : make it ascii  
 00000000'EF FAD0 CF 00 0924 430 : MOVL RECCNT, CPYBUF : insert rec #  
 2D 57 6E 00 2C 092D 431 : MOVC5 #0, (SP), R7, #45, CPYBUF+4 : fill rec with char  
 00000004'EF 0932 432 : SPUT RAB=R11, ERR=REPORT\_ERROR  
 F6B7' 30 0946 433 : BSBW ERR  
 05A1' 30 0949 434 : BSBW CHKRFA  
 094C 435 :  
 094C 436 :  
 094C 437 : print message every 100 records

RMSTEST1  
V04-000

GENERAL RMS TEST PROGRAM :

J 5

16-SEP-1984 01:45:37 VAX/VMS Macro V04-00  
5-SEP-1984 04:21:39 [UETP.SRC]RMSTEST1.MAR;1

Page 10  
(10)

094C 438 :  
52 FAA3 CF 00000064 8F 7B 094C 439  
57 0950 440 EDIV #100,RECCNT,R2,R7  
57 0957 441 TSTL R7  
03 12 0959 442 BNEQ T2CNT  
0537 30 095B 443 BSBW TYPFRA  
095E 444  
095E 445 :  
095E 446  
FFB2 FA92 CF 01 03E8 8F 3D 095E 447 T2CNT: ACBW #1000 #1,RECCNT,NXTRC2  
F686' 30 0977 448 SCLOSE FAB=R10,ERR=REPORT\_ERROR  
02 AB B4 097A 450 BSBW ERR  
097D 451 CLRW RABSW ISI(R11)  
WTTYPE <END STEP 2A - START STEP 2B>

RM  
V0

09B9 453  
 09B9 454  
 09B9 455 : test 2b  
 09B9 456  
 09B9 457 : reread file created in step 2a and try random and sequential  
 09B9 458 : access via \$get  
 09B9 459 :  
 09B9 460  
 04 AA 16 AA 09 90 09B9 461 MOVBL #FABSM\_PUT!FABSM\_UPD,FABSB\_FAC(R10); upd implies get access  
 01000000 8F C8 09BD 462 BISL2 #FABSM\_NAM,FABSL\_FOP(R10)  
 F629' 30 09C5 463 \$OPEN FAB=R10,ERR=REPORT\_ERROR  
 09D4 464 BSBW ERR  
 09D7 465  
 09D7 466 :  
 09D7 467 : movl #rabSm\_loc,rab\$1\_rop(r11)  
 09D7 468 :  
 09D7 469 :  
 1E AB 01 90 09D7 470 \$RAB\_STORE RAB=(R11),ROP=<LOC,UIF>  
 F9FC CF 03E8 8F F60B' 30 09DF 471 MOVB #RABSC\_KEY,RABSB\_RAC(R11)  
 03E8 8F 3C 09E3 472 \$CONNECT RAB=R11,ERR=REPORT\_ERROR  
 09F2 473 BSBW ERR  
 09F5 474 MOVZWL #1000,RECCNT  
 09FC 475  
 09FC 476 : get all records in reverse order  
 09FC 477 :  
 09FC 478 :  
 09FC 479 :  
 FFF3 F9F1 CF FFFF 8F 054D 30 09FC 480 10\$: BSBW GTCHK2  
 02 3D 09FF 481 ACBW #2,#-1,RECCNT,10\$  
 0A09 482  
 0A09 483 : now get them all forward  
 0A09 484 :  
 0A09 485 :  
 0A09 486 :  
 FFF3 F9E4 CF 01 03E8 8F 0540 30 0A09 487 20\$: BSBW GTCHK2  
 3D 0A0C 488 ACBW #1000,#1,RECCNT,20\$  
 0A16 489  
 0A16 490 : now get every 10th record in reverse order followed by the  
 0A16 491 : next 10 in sequential order  
 0A16 492 :  
 0A16 493 :  
 0A16 494 :  
 F9DB CF 03DE 8F 80 0A16 495 30\$: MOVW #990,RECCNT  
 052C 30 0A1D 496 BSBW GTCHK2  
 1E AB 00 90 0A20 497 MOVB #RABSC\_SEQ,RABSB\_RAC(R11)  
 58 0A D0 0A24 498 MOVL #10,RB  
 F9CD CF D6 0A27 499 25\$: INCL RECCNT  
 051E 30 0A2B 500 BSBW GTCHK2  
 F6 58 F5 0A2E 501 SOBGTR R8,25\$  
 FFDE F9BB CF 1E AB 01 90 0A31 502 MOVB #RABSC\_KEY,RABSB\_RAC(R11)  
 FFEC 8F 01 3D 0A35 503 ACBW #1,#-20,RECCNT,30\$  
 0A3F 504  
 0A3F 505 :  
 0A3F 506 :  
 0A3F 507 :  
 WTTYPE <PASS 1 O.K.>

0A7B 509  
 0A7B 510 :  
 0A7B 511 : do 10 random puts, changing record contents  
 0A7B 512 :  
 0A7B 513 :  
 0A7B 514 :  
 0A7E 515 :  
 0A84 516 40\$: MOVB #^A/0/ R6 ; updating character  
 0A8D 517 : MOVZBL #91,RECCNT  
 0A92 518 : MOVL RECCNT,CPYBUF  
 0A97 519 : MOVCS #0,(SP),R6,#45,CPYBUF+4 ; change the record  
 0AA6 520 :  
 0AA9 521 : SPUT RAB=R11,ERR=REPORT\_ERROR  
 0AAC 522 : BSBW ERR  
 0AB6 523 : BSBW CHKRFA  
 0AB8 524 : ACBW #1000,#101,RECCNT,40\$  
 0ABB 525 :  
 0ABB 526 : BSBW CHKMOD ; go verify changes  
 0ABB 527 :  
 0ABB 528 :  
 0ABB 529 :  
 0ABE 530 50\$: MOVB #^A/1/ R6 ; updating character  
 0AC4 531 : MOVZBL #91,RECCNT ; starting rec #  
 0AD3 532 : SGET RAB=R11,ERR=REPORT\_ERROR  
 0AD6 533 : BSBW ERR  
 0ADD 534 55\$: MOVAL CPYBUF+4,R8 ; get addr of record char  
 0AE0 535 : MOVL #45,R9  
 0AE2 536 : INCB (R8)+ ; bump contents  
 0AE5 537 : SOBGTR R9,55\$  
 0AF4 538 : \$UPDATE RAB=R11,ERR=REPORT\_ERROR  
 0AF7 539 : BSBW ERR  
 0AF8 540 : BSBW CHKRFA  
 0B04 541 : ACBW #1000,#101,RECCNT,50\$  
 0B06 542 :  
 0B09 543 :  
 0B09 544 :  
 0B09 545 :  
 0B09 546 :  
 0B09 547 :  
 0B09 548 :  
 0B09 549 :  
 0B09 550 :  
 0B09 551 :  
 0B09 552 :  
 0B09 553 :  
 0B09 554 :  
 0B09 555 :  
 0B09 556 :  
 0B09 557 :  
 0B09 558 :  
 0B09 559 :  
 0B09 560 :  
 0B09 561 :  
 0B09 562 :  
 F974 CF 56 30 90 0A7B 509 :  
 00000000'EF 5B 8F 9A 0A7E 510 :  
 2D 56 6E 00 2C 0A84 511 : do 10 random puts, changing record contents  
 00000004'EF 0A8D 512 :  
 F557' 30 0AA6 513 :  
 0441 30 0AA9 514 :  
 CC F942 CF 0065 8F 03E8 8F 3D 0AAC 515 :  
 06EF 30 0AB6 516 :  
 0AB8 517 :  
 0ABB 518 :  
 0ABB 519 :  
 0ABB 520 :  
 0ABB 521 :  
 0ABB 522 :  
 0ABB 523 :  
 0ABB 524 :  
 0ABB 525 :  
 0ABB 526 :  
 0ABB 527 :  
 F934 CF 56 31 90 0ABB 528 :  
 5B 8F 9A 0ABE 529 :  
 0AC4 530 50\$: MOVB #^A/1/ R6 ; updating character  
 58 00000004'EF 531 : MOVZBL #91,RECCNT ; starting rec #  
 F52A' 30 0AD3 532 : SGET RAB=R11,ERR=REPORT\_ERROR  
 59 2D 0AD6 533 : BSBW ERR  
 0ADD 534 55\$: MOVAL CPYBUF+4,R8 ; get addr of record char  
 88 96 0AE0 535 : MOVL #45,R9  
 FB 59 F5 0AE2 536 : INCB (R8)+ ; bump contents  
 F509' 30 0AE5 537 : SOBGTR R9,55\$  
 03F3 30 0AF4 538 : \$UPDATE RAB=R11,ERR=REPORT\_ERROR  
 BE F8F4 CF 0065 8F 03E8 8F 3D 0AF7 539 :  
 FF 0B04 540 : BSBW ERR  
 0B06 541 : BSBW CHKRFA  
 0B09 542 : ACBW #1000,#101,RECCNT,50\$  
 0B09 543 :  
 0B09 544 :  
 0B09 545 :  
 0B09 546 :  
 0B09 547 :  
 0B09 548 :  
 0B09 549 :  
 0B09 550 :  
 0B09 551 :  
 0B09 552 :  
 0B09 553 :  
 0B09 554 :  
 0B09 555 :  
 0B09 556 :  
 0B09 557 :  
 0B09 558 :  
 0B09 559 :  
 0B09 560 :  
 0B09 561 :  
 0B09 562 :  
 F4A9' 30 0B54 544 :  
 3F 1E AA 03 E2 0B57 545 :  
 FCDB 31 0B5C 546 :  
 0B54 547 :  
 0B57 548 :  
 0B5C 549 :  
 0B98 550 :  
 0B98 551 :  
 0B98 552 :  
 16 AA 01 90 0B98 553 :  
 1E AB 00 90 0B9F 554 :  
 00000004'EF 00000400 8F DO 0BA3 555 :  
 F7A4 CF 01000000 8F CA 0B80 556 :  
 0B80 557 :  
 0B8B 558 :  
 0BC4 559 :  
 0BC4 560 :  
 0BC4 561 :  
 0BC4 562 :  
 0B09 563 :  
 0B09 564 :  
 0B09 565 :  
 0B09 566 :  
 0B09 567 :  
 0B09 568 :  
 0B09 569 :  
 0B09 570 :  
 0B09 571 :  
 0B09 572 :  
 0B09 573 :  
 0B09 574 :  
 0B09 575 :  
 0B09 576 :  
 0B09 577 :  
 0B09 578 :  
 0B09 579 :  
 0B09 580 :  
 0B09 581 :  
 0B09 582 :  
 0B09 583 :  
 0B09 584 :  
 0B09 585 :  
 0B09 586 :  
 0B09 587 :  
 0B09 588 :  
 0B09 589 :  
 0B09 590 :  
 0B09 591 :  
 0B09 592 :  
 0B09 593 :  
 0B09 594 :  
 0B09 595 :  
 0B09 596 :  
 0B09 597 :  
 0B09 598 :  
 0B09 599 :  
 0B09 600 :  
 0B09 601 :  
 0B09 602 :  
 0B09 603 :  
 0B09 604 :  
 0B09 605 :  
 0B09 606 :  
 0B09 607 :  
 0B09 608 :  
 0B09 609 :  
 0B09 610 :  
 0B09 611 :  
 0B09 612 :  
 0B09 613 :  
 0B09 614 :  
 0B09 615 :  
 0B09 616 :  
 0B09 617 :  
 0B09 618 :  
 0B09 619 :  
 0B09 620 :  
 0B09 621 :  
 0B09 622 :  
 0B09 623 :  
 0B09 624 :  
 0B09 625 :  
 0B09 626 :  
 0B09 627 :  
 0B09 628 :  
 0B09 629 :  
 0B09 630 :  
 0B09 631 :  
 0B09 632 :  
 0B09 633 :  
 0B09 634 :  
 0B09 635 :  
 0B09 636 :  
 0B09 637 :  
 0B09 638 :  
 0B09 639 :  
 0B09 640 :  
 0B09 641 :  
 0B09 642 :  
 0B09 643 :  
 0B09 644 :  
 0B09 645 :  
 0B09 646 :  
 0B09 647 :  
 0B09 648 :  
 0B09 649 :  
 0B09 650 :  
 0B09 651 :  
 0B09 652 :  
 0B09 653 :  
 0B09 654 :  
 0B09 655 :  
 0B09 656 :  
 0B09 657 :  
 0B09 658 :  
 0B09 659 :  
 0B09 660 :  
 0B09 661 :  
 0B09 662 :  
 0B09 663 :  
 0B09 664 :  
 0B09 665 :  
 0B09 666 :  
 0B09 667 :  
 0B09 668 :  
 0B09 669 :  
 0B09 670 :  
 0B09 671 :  
 0B09 672 :  
 0B09 673 :  
 0B09 674 :  
 0B09 675 :  
 0B09 676 :  
 0B09 677 :  
 0B09 678 :  
 0B09 679 :  
 0B09 680 :  
 0B09 681 :  
 0B09 682 :  
 0B09 683 :  
 0B09 684 :  
 0B09 685 :  
 0B09 686 :  
 0B09 687 :  
 0B09 688 :  
 0B09 689 :  
 0B09 690 :  
 0B09 691 :  
 0B09 692 :  
 0B09 693 :  
 0B09 694 :  
 0B09 695 :  
 0B09 696 :  
 0B09 697 :  
 0B09 698 :  
 0B09 699 :  
 0B09 700 :  
 0B09 701 :  
 0B09 702 :  
 0B09 703 :  
 0B09 704 :  
 0B09 705 :  
 0B09 706 :  
 0B09 707 :  
 0B09 708 :  
 0B09 709 :  
 0B09 710 :  
 0B09 711 :  
 0B09 712 :  
 0B09 713 :  
 0B09 714 :  
 0B09 715 :  
 0B09 716 :  
 0B09 717 :  
 0B09 718 :  
 0B09 719 :  
 0B09 720 :  
 0B09 721 :  
 0B09 722 :  
 0B09 723 :  
 0B09 724 :  
 0B09 725 :  
 0B09 726 :  
 0B09 727 :  
 0B09 728 :  
 0B09 729 :  
 0B09 730 :  
 0B09 731 :  
 0B09 732 :  
 0B09 733 :  
 0B09 734 :  
 0B09 735 :  
 0B09 736 :  
 0B09 737 :  
 0B09 738 :  
 0B09 739 :  
 0B09 740 :  
 0B09 741 :  
 0B09 742 :  
 0B09 743 :  
 0B09 744 :  
 0B09 745 :  
 0B09 746 :  
 0B09 747 :  
 0B09 748 :  
 0B09 749 :  
 0B09 750 :  
 0B09 751 :  
 0B09 752 :  
 0B09 753 :  
 0B09 754 :  
 0B09 755 :  
 0B09 756 :  
 0B09 757 :  
 0B09 758 :  
 0B09 759 :  
 0B09 760 :  
 0B09 761 :  
 0B09 762 :  
 0B09 763 :  
 0B09 764 :  
 0B09 765 :  
 0B09 766 :  
 0B09 767 :  
 0B09 768 :  
 0B09 769 :  
 0B09 770 :  
 0B09 771 :  
 0B09 772 :  
 0B09 773 :  
 0B09 774 :  
 0B09 775 :  
 0B09 776 :  
 0B09 777 :  
 0B09 778 :  
 0B09 779 :  
 0B09 780 :  
 0B09 781 :  
 0B09 782 :  
 0B09 783 :  
 0B09 784 :  
 0B09 785 :  
 0B09 786 :  
 0B09 787 :  
 0B09 788 :  
 0B09 789 :  
 0B09 790 :  
 0B09 791 :  
 0B09 792 :  
 0B09 793 :  
 0B09 794 :  
 0B09 795 :  
 0B09 796 :  
 0B09 797 :  
 0B09 798 :  
 0B09 799 :  
 0B09 800 :  
 0B09 801 :  
 0B09 802 :  
 0B09 803 :  
 0B09 804 :  
 0B09 805 :  
 0B09 806 :  
 0B09 807 :  
 0B09 808 :  
 0B09 809 :  
 0B09 810 :  
 0B09 811 :  
 0B09 812 :  
 0B09 813 :  
 0B09 814 :  
 0B09 815 :  
 0B09 816 :  
 0B09 817 :  
 0B09 818 :  
 0B09 819 :  
 0B09 820 :  
 0B09 821 :  
 0B09 822 :  
 0B09 823 :  
 0B09 824 :  
 0B09 825 :  
 0B09 826 :  
 0B09 827 :  
 0B09 828 :  
 0B09 829 :  
 0B09 830 :  
 0B09 831 :  
 0B09 832 :  
 0B09 833 :  
 0B09 834 :  
 0B09 835 :  
 0B09 836 :  
 0B09 837 :  
 0B09 838 :  
 0B09 839 :  
 0B09 840 :  
 0B09 841 :  
 0B09 842 :  
 0B09 843 :  
 0B09 844 :  
 0B09 845 :  
 0B09 846 :  
 0B09 847 :  
 0B09 848 :  
 0B09 849 :  
 0B09 850 :  
 0B09 851 :  
 0B09 852 :  
 0B09 853 :  
 0B09 854 :  
 0B09 855 :  
 0B09 856 :  
 0B09 857 :  
 0B09 858 :  
 0B09 859 :  
 0B09 860 :  
 0B09 861 :  
 0B09 862 :  
 0B09 863 :  
 0B09 864 :  
 0B09 865 :  
 0B09 866 :  
 0B09 867 :  
 0B09 868 :  
 0B09 869 :  
 0B09 870 :  
 0B09 871 :  
 0B09 872 :  
 0B09 873 :  
 0B09 874 :  
 0B09 875 :  
 0B09 876 :  
 0B09 877 :  
 0B09 878 :  
 0B09 879 :  
 0B09 880 :  
 0B09 881 :  
 0B09 882 :  
 0B09 883 :  
 0B09 884 :  
 0B09 885 :  
 0B09 886 :  
 0B09 887 :  
 0B09 888 :  
 0B09 889 :  
 0B09 890 :  
 0B09 891 :  
 0B09 892 :  
 0B09 893 :  
 0B09 894 :  
 0B09 895 :  
 0B09 896 :  
 0B09 897 :  
 0B09 898 :  
 0B09 899 :  
 0B09 900 :  
 0B09 901 :  
 0B09 902 :  
 0B09 903 :  
 0B09 904 :  
 0B09 905 :  
 0B09 906 :  
 0B09 907 :  
 0B09 908 :  
 0B09 909 :  
 0B09 910 :  
 0B09 911 :  
 0B09 912 :  
 0B09 913 :  
 0B09 914 :  
 0B09 915 :  
 0B09 916 :  
 0B09 917 :  
 0B09 918 :  
 0B09 919 :  
 0B09 920 :  
 0B09 921 :  
 0B09 922 :  
 0B09 923 :  
 0B09 924 :  
 0B09 925 :  
 0B09

		0BC4	563	:	test truncate
		0BC4	564	:	
		0BC4	565	:	
		0BC4	566		\$FAB STORE FAB=R10,SHR=<PUT,GET,TRN>; set up for trn
		0BCB	567		\$OPEN FAB=R10,ERR=REPORT_ERROR
F423'	30	0BDA	568		BSBW ERR
F411'	30	0BEC	569		\$CONNECT RAB=R11,ERR=REPORT_ERROR
F3FF'	30	0BEF	570		BSBW ERR
F3ED'	30	0BFE	571		\$GET RAB=R11,ERR=REPORT_ERROR; get 1st record, setting up cd
		0C01	572		BSBW ERR
		0C10	573		\$TRUNCATE RAB=R11,ERR=REPORT_ERROR; truncate all
		0C13	574		BSBW ERR
0001827A	BF	D1	0C1C	575	\$GET RAB=R11 ; should get eof
		06	0C23	576	CMPL R0,#RMSS_EOF
5A	5B	D0	0C25	577	BEQL TR0K
F3D5'	30	0C28	578		MOVL R11,R10
		0C2B	579		BSBW EOFPUT
		0C5A	580	TROK:	TYPE <TRUNCATE SUCCEEDED!> ; too bad
F394'	30	0C69	581		\$DISCONNECT RAB=R11,ERR=REPORT_ERROR; clean up
F382'	30	0C6C	582		BSBW ERR
		0C7B	583		\$CLOSE FAB=R10,ERR=REPORT_ERROR
		0C7E	584		BSBW ERR
36	1F AA 03	90	0C87	585	\$FAB_STORE FAB=R10,SHR=<PUT,GET,UPI>,FAC=PUT
	AA 64 8F	98	0C8B	586	MOVB #FABSC VFC,FABSB RFM(R10)
	1E AA 02	90	0C90	587	MOVZBW #100,FABSW MRS(RT0)
	10 AA 30	D0	0C94	588	MOVB #FABSM CR,FABSB RAT(R10)
	3F AA 04	90	0C98	589	MOVL #4,FABSL ALQ(RT0)
28 AB	00000000'EF	DE	0C9C	590	MOVB #4,FABSB FSZ(R10)
				591	MOVAL CPYBUF,RABSL_RBF(R11)
		0CA4	592		
		0CA4	593	:	
		0CA4	594	:	try to erase the file!
		0CA4	595	:	
		0CA4	596		
F348'	30	0CA4	597		SERASE FAB=T1FAB,ERR=REPORT_ERROR
		0CB5	598		BSBW ERR
F678 CF	00100000 8F	C8	0CB8	599	TYPE <T1FILE.DAT HAS BEEN ERASED>
		0CE7	600		BISL2 #FABSM CTG,FAB\$L FOP+T1FAB; for 'f' test
		0CF0	601		FINISH <SEQUENTIAL TESTS>
		04	0D05	602	RET

			0D06	604			
			0D06	605	;		
			0D06	606	subroutine to read in a record		
			0D06	607	;		
			0D06	608			
			0D06	609	GETANDCHK:		
10 AB	68 06	28	0D06	610	MOVC3	#6,(R8),RABSW_RFA(R11)	: rfa to rab
15 57	E8	0D08	611	BLBS	R7,PASS2	branch if pass 2	
F2E0'	30	0D1D	612	\$GET	RAB=R11,ERR=REPORT_ERROR	get via rfa	
3C	10	0D20	613	BSBW	ERR		
05	05	0D22	614	BSBB	CHKREC		
			615	RSB			
1E AB	02	90	0D23	616	PASS2:		
			0D27	617	MOVB	#RABSC_RFA,RABSB_RAC(R11)	
F2C7'	30	0D36	618	\$FIND	RAB=R1T,ERR=REPORT_ERROR		
1E AB	00	90	0D39	619	BSBW	ERR	
0A	DD	0D3D	620	MOVB	#RABSC_SEQ,RABSB_RAC(R11);	switch back to sequential	
		0D3F	621	PUSHL	#10	; loop count	
			622	T1BLOOP3:			
F2AF'	30	0D4E	623	\$GET	RAB=R11,ERR=REPORT_ERROR		
0B	10	0D51	624	BSBW	ERR		
59	D6	0D53	625	BSBB	CHKREC		
E7	6E	F5	0D55	626	INCL	R9	: bump record count
59	0A	C2	0D58	627	SORGTR	(SP),T1BLOOP3	
	8E	D5	0D5B	628	SUBL2	#10,R9	: restore record count
	05	05	0D5D	629	TSTL	(SP)+	: clean up stack
			630	RSB			

OD5E 632  
OD5E 633 :  
OD5E 634 : subroutine to check that record read is really the right record  
OD5E 635 :  
OD5E 636 :  
OD5E 637 :  
F695 CF 59 D1 OD5F 638 :  
44 12 OD63 639 :  
56 52 59 00000064 8F 7B OD65 640 :  
22 AB 56 B1 OD6E 641 :  
55 52 F67F CF 0A 7B OD74 642 :  
55 30 28 BB 56 80 OD7B 643 :  
6E 00 55 28 BB 56 2D OD7E 644 :  
50 12 OD85 645 :  
10 AB 68 07 57 E8 OD87 646 :  
06 29 OD8A 648 :  
01 12 OD8F 649 :  
05 OD91 650 10\$: RSB  
CHKREC: CMPL R9, RECCNT ; fixed header = record #?  
BNEQ BADRHB  
CHKRC1: EDIV #100, R9, R2, R6 ; compute record length  
CMPW R6, RABSW\_RSZ(R11) ; = rms record len?  
BNEQ BADRSZ  
EDIV #10, RECCNT, R2, R5 ; compute character  
ADD8 #48, R5 ; make ascii  
CMPCS R6, RABSL\_RBF(R11), R5, #0, (SP); match the record?  
BNEQ BADRBF  
BLBS R7, 10\$ ; branch if pass 2  
CMPC3 #6, (R8), RABSW\_RFA(R11) ; rfa the same?  
BNEQ BADRFA

```

0D92 652
0D92 653
0D92 654 : handle errors
0D92 655
0D92 656
46 11 0D92 657 BADRFA: FIELD <RFA>
        0DA7 658 BRB   ERROR
        0DA9 659 BADRHB: FIELD <RHB>
        0DBE 660 BRB   ERROR
        0DC0 661 BADRSZ: FIELD <RSZ>
18 11 0DD5 662 BRB   ERROR
        0DD7 663 BADRBF: FIELD <RBF>
01 11 0DEC 664 BRB   ERROR
        0DEE 665
        0DEE 666 :
        0DEE 667
        05 0DEF 668 DONT: RSB
        0DEF 669 ERROR:
F8 00000000'EF E9 0DEF 670 BLBC  VERBOSITY,DONT
00000004'EF 01 CA 0DF6 671 SWAIT RAB=CMDORAB
        0E03 672 BICL2 #RABSM ASY,RABSL ROP+CMDORAB
        0EOA 673 SFAO S T2STR,CMDORAB+RAB$W_RSZ,FAOBUF,-
        0EOA 674 R9 RECCNT,R6,RAB$W_RSZ(R11),-
        0EOA 675 (R8),4(R8),RABSW_RFA(R11),RABSW_RFA+4(R11)
00000028'EF 00000000'EF 9E 0E37 676 BLBC  R0,
        0E3A 677 MOVAB CMDBUF,CMDORAB+RABSL RBF
        0E45 678 SPUT  RAB=CMDORAB,ERR=REPORT_ERROR
50 22 AB F1A5' 30 0E58 679 BSBW  ERR
        0E5B 680 MOVZWL RABSW_RSZ(R11),R0
        0E5F 681 SFAO S T3STR,CMDORAB+RAB$W_RSZ,FAOBUF,-
        0E5F 682 R0,RABSL RBF(R11)
        FF 50 E9 0E7B 683 BLBC  R0,
        0E7E 684 SPUT  RAB=CMDORAB,ERR=REPORT_ERROR
        F16C' 30 0E91 685 BSBW  ERR
        05 0E94 686 RSB
        0E95 687
        0E95 688
        0E95 689 : output a message
        0E95 690
        0E95 691
50 00000000'EF E9 0E95 692 TYPRFA: BLBC  VERBOSITY,NO
        0E9C 693 SWAIT RAB=WTRAB
        0EA7 694 SFAO S T1STR,CMDORAB+RAB$W_RSZ,FAOBUF,RECCNT,RABSW_RFA(R11),-
        0EA7 695 RAB$W_RFA+4(R11)
        FF 50 E9 0EC8 696 BLBC  R0,
00000028'EF 00000000'EF 9E 0ECB 697 MOVAB CMDBUF,CMDORAB+RABSL RBF
        0ED6 698 SPUT  RAB=CMDORAB,ERR=REPORT_ERROR
        F114' 30 0EE9 699 BSBW  ERR
        05 0EEC 700 NO: RSB

```

			OEED	702			
			OEED	703			
			OEED	704			subroutines for test 2
			OEED	705			
			OEED	706			subroutine to check rfa correctness
			OEED	707			
			OEED	708			
50	F506	CF	01	C3	OEED	709	CHKRFA: SUBL3 #1,RECCNT,R0
51	10	1E	AA	03	E0	710	BBS #FAB\$V_BLK,FAB\$B_RAT(R10),10\$
52	51	00	32	50	7A	711	EMUL R0,#50,#0,R1
					OEFD	712	EDIV #512,R1,R1,R2
					OF06	713	BRB 20\$
					OF08	714	
					OF08	715	
					OF08	716	alternate calc for records not crossing blk boundaries
					OF08	717	
					OF08	718	
52	51	50	51	D4	OF08	719	10\$: CLRL R1
			52	0A	7B	720	EDIV #10,R0,R1,R2
				32	OF0A	721	MULW2 #50,R2
				51	A4	722	20\$: INCL R1
10	AB	51	D6	OF12	723	723	CMPL R1,RABSW_RFA(R11)
		07	D1	OF14	724		BNEQ ERRRFA
14	AB	52	07	12	OF18	725	CMPW R2,RABSW_RFA+4(R11)
		01	52	B1	OF1A	726	BNEQ ERRRFA
			01	12	OF1E	727	RSB
				05	OF20	728	OF21 ERRRFA: WFIELD <RFA>
					OF43	729	MBPT
				05	OF4B	730	RSB

```

        OF4C 732
        OF4C 733
        OF4C 734 : subroutine to get and check a record for test 2
        OF4C 735
        OF4C 736
        OF4C 737 GTCHK2: EDIV #42,RECCNT,R0,R7 ; compute char
        OF53 738 ADDB #48,R7
        OF56 739 GTCHK2A:
        OF56 740 $GET RAB=R11,ERR=REPORT_ERROR
        OF65 741 BSBW ERR
        OF68 742 BSBW CHKRFA
        OF6B 743 MOVL RABSL_RBF(R11),R2 ; get rec addr
        OF6F 744 CMPL (R2)+,RECCNT ; recnt o.k.?
        OF74 745 BNEQ ERRREC
        OF76 746 CMPCS #45,(R2),R7,#0,(SP) ; match?
        OF7C 747 BNEQ ERRREC
        OF7E 748 RSB
        OF7F 749 ERRREC: WFIELD <RECORD CONTENTS>
        OFA1 750 MBPT
        OFA9 751 RSB
        OFAA 752
        OFAA 753 : subroutine to verify that updated records were changed but not
        OFAA 754 : the preceeding or following records
        OFAA 755 :
        OFAA 756 :
        OFAA 757 :
        OFAA 758 CHKMOD:
        F448 CF 5A 8F 9A OFAA 759 MOVZBL #90,RECCNT ; starting rec #
        FF99 30 OFB0 760 CHKNXT: BSBW GTCHK2
        F441 CF D6 OFB3 761 INCL RECCNT
        57 56 00 OFB7 762 MOVL R6,R7 ; check character
        FF99 30 OFBA 763 BSBW GTCHK2A
        F437 CF D6 OFBD 764 INCL RECCNT
        F433 CF B1 OFC1 765 CMPW RECCNT,#1001 ; all done?
        0C 13 OFC8 766 BEQL 10S
        FF7F 30 OFCA 767 BSBW GTCHK2
        F424 CF 0063 8F A0 OFCD 768 ADDW #99,RECCNT
        DA 11 OFD4 769 BRB CHKNXT
        0001827A 8F 50 D1 OFDF 770 10S: $GET R11
        01 12 OFE6 771 CMPL R0,#RMSS_EOF
        05 OFE8 772 BNEQ 20S
        OFE9 773 RSB
        OFE9 774 20S: $WAIT RAB=CMDORAB
        5A 5A DD OFF6 775 PUSHL R10
        5B 5B 00 OFF8 777 MOVL R11,R10 ; save it
        F002 30 OFFB 778 BSBW EOFPUT ; bad structure!!!
        0400 8F BA OFFE 779 POPR #^M<R10>
        1002 100A 780 MBPT
        1008 781 RSB
        1008 782
        1008 783 .END

```

\$\$PSECT_EP	= 00000000		FABSM_CTG	= 00100000	D
\$\$TAB	= 000003B4	R D 01	FABSM_GET	= 00000002	D
\$\$TABEND	= 000003F8	R D 01	FABSM_NAM	= 01000000	D
\$\$TMP	= 00000043		FABSM_PUT	= 00000001	D
\$\$TMP1	= 00000001	D	FABSM_UPD	= 00000008	D
\$\$TMP2	= 000000EF		FABSV_BLK	= 00000003	D
\$\$TMPX	= 00000187	R D 04	FABSV_CHAN_MODE	= 00000002	D
\$\$TMPX1	= 0000000F	D	FABSV_CR	= 00000001	D
\$\$RMSTEST	= 0000001E		FABSV_CTG	= 00000014	D
\$\$RMS_PBUGCHK	= 00000010		FABSV_FILE_MODE	= 00000004	D
\$\$RMS_TBUGCHK	= 00000008		FABSV_GET	= 00000001	D
\$\$RMS_UMODE	= 00000004		FABSV_LNM_MODE	= 00000000	D
\$\$T2	= 00000006	D	FABSV_PUT	= 00000000	D
..AFLG	= 00000000		FABSV_SUP	= 00000002	D
..FLG	= 00000002		FABSV_TRN	= 00000004	D
..MOD	= 00000001		FABSVUPI	= 00000006	D
..N	= 00000003		FABSW_GBC	= 00000048	D
..TYP	= 00000003		FABSW_MRS	= 00000036	D
.LEN	= 00000001		FAOBUF	*****	X 01
BADRBF	00000DD7	R D 01	FINPUT	*****	X 01
BADRFA	00000D92	RR D 01	FIN_DESCR	*****	X 01
BADRHB	00000DA9	RR D 01	FLDPUT	*****	X 01
BADRSZ	00000DC0	RR D 01	FLD_DESCR	*****	X 01
BEGPUT	*****	X 01	GETANDCHK	00000D06	R D 01
BEG_DESCR	*****	X 01	GTCHK2	00000F4C	R D 01
CHKMOD	00000FAA	R D 01	GTCHK2A	00000F56	R D 01
CHKNXT	00000FB0	RR D 01	NAMBLK	*****	X 01
CHKRC1	00000D65	RR D 01	NO	00000EEC	R D 01
CHKREC	00000D5E	RR D 01	NXTRC2	0000091A	R D 01
CHKRFA	00000EED	R D 01	NXTREC	00000511	R D 01
CMDBUF	*****	X 01	PASS2	00000D23	R D 01
CMDORAB	*****	X 01	RAB\$B_RAC	= 000001E	D
CPYBSZ	*****	X 01	RAB\$C_BID	= 0000001	D
CPYBUF	*****	X 01	RAB\$C_BLN	= 0000044	D
DONE	00000B9B	R D 01	RAB\$C_KEY	= 0000001	D
DONT	00000DEE	R D 01	RAB\$C_RFA	= 0000002	D
EOFPUT	*****	X 01	RAB\$C_SEQ	= 0000000	D
ERR	*****	X 01	RAB\$L_CTX	= 0000018	D
ERROR	00000DEF	R D 01	RAB\$L_RBF	= 0000028	D
ERRREC	00000F7F	RR D 01	RAB\$L_RHB	= 000002C	D
ERRRFA	00000F21	R D 01	RAB\$L_ROP	= 0000004	D
FAB\$B_FAC	== 00000016		RABSM_ASY	= 0000001	D
FAB\$B_FNS	== 00000034		RABSM_LOC	= 0010000	D
FAB\$B_FSZ	== 0000003F		RABSM_WBH	= 0000400	D
FAB\$B_RAT	== 0000001E		RABSV_LOC	= 0000010	D
FAB\$B_RFM	== 0000001F		RABSV_UIF	= 0000004	D
FAB\$B_SHR	== 00000017		RABSV_WBH	= 000000A	D
FAB\$C_BID	== 00000003		RABSW_ISI	= 0000002	D
FAB\$C_BLN	== 00000050		RABSW_RFA	= 0000010	D
FAB\$C_FIX	== 00000001		RABSW_RSZ	= 00000022	D
FAB\$C_SEQ	== 00000000		RECCNT	000003F8	RG D 01
FAB\$C_VFC	== 00000003		REPORT_ERROR	*****	X 01
FAB\$L_ALQ	== 00000010		RFATBL	00000044	R D 01
FAB\$L_FNA	== 0000002C		RHBSW	000004AB	R D 01
FAB\$L_FOP	== 00000004		RMSS_EOF	0001827A	D
FAB\$L_STS	== 00000008		RMSS_SUPERSEDE	00010631	D
FABSM_CR	== 00000002	D	RMT\$TEST_1A	000004AC	RG D 01

SUPOK	000008DB	R D	01
SYSSCLOSE	*****	GX	01
SYSSCONNECT	*****	GX	01
SYSSCREATE	*****	GX	01
SYSSDISCONNECT	*****	GX	01
SYSSERASE	*****	GX	01
SYSSFAO	*****	X	01
SYSSFIND	*****	GX	01
SYSSGET	*****	GX	01
SYSSOPEN	*****	GX	01
SYSSPUT	*****	GX	01
SYSSREWIND	*****	GX	01
SYSSTRUNCATE	*****	GX	01
SYSSUPDATE	*****	GX	01
SYSSWAIT	*****	GX	01
T1BDONE	000006A1	R D	01
T1BLOOP1	0000063B	R D	01
T1BLOOP2	0000064C	R D	01
T1BLOOP3	00000D3F	R D	01
T1BLOOP4	00000769	R D	01
T1BLOOP5	000007C9	R D	01
T1CNT	00000556	R D	01
T1FAB	00000364	RG D	01
T1L	= 00000025	D	
T1RAB	000003B4	RG D	01
T1S	00000408	R D	01
T1START	00000000	RG D	01
T1STR	00000400	R D	01
T2CNT	= 0000095E	R D	01
T2L	= 0000005F	D	
T2S	00000435	R D	01
T2SETUP	00000876	R D	01
T2STR	= 0000042D	R D	01
T3L	= 0000000F	D	
T3S	0000049C	R D	01
T3STR	00000494	R D	01
TROK	00000C2B	R D	01
TYPRFA	00000E95	R D	01
VERBOSITY	*****	X	01
WTRAB	00000000	RG D	01

-----+  
! Psect synopsis !  
-----+

## PSECT name

-----	-----	-----	-----											
. ABS	00000000	( 0.)	00 ( 0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
RMSTEST	0000100B	( 4107.)	01 ( 1.)	NOPIC	USR	CON	REL	GBL	NOSHR	EXE	RD	WRT	NOVEC	LONG
\$_ABS	00000000	( 0.)	02 ( 2.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
\$_RMSNAM	00000015	( 21.)	03 ( 3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
\$_RMSNAM	00000196	( 406.)	04 ( 4.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

## Allocation

## PSECT No.

## Attributes

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	38	00:00:00.07	00:00:00.55
Command processing	133	00:00:00.54	00:00:02.18
Pass 1	376	00:00:17.01	00:00:35.60
Symbol table sort	0	00:00:00.74	00:00:01.42
Pass 2	138	00:00:03.96	00:00:06.21
Symbol table output	18	00:00:00.13	00:00:00.17
Psect synopsis output	2	00:00:00.03	00:00:00.12
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	707	00:00:22.49	00:00:46.25

The working set limit was 1650 pages.

86298 bytes (169 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 618 non-local and 34 local symbols.

783 source lines were read in Pass 1, producing 58 object records in Pass 2.

69 pages of virtual memory were used to define 50 macros.

PSE  
---  
-  
SAE  
SRI  
---

Pha  
---  
In  
Co  
Pas  
Syn  
Pas  
Syn  
Pas  
Crc  
Ass

The  
472  
The  
440  
38

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	36
TOTALS (all libraries)	36

1074 GETS were required to define 36 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:RMSTEST1/OBJ=OBJ\$:RMSTEST1 MSRC\$:RMSTEST1/UPDATE=(ENH\$:RMSTEST1)+EXECML\$/LIB

Mac  
---  
-  
SA  
T01  
791  
The  
MAI

0409 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

